ECOGRAPHY

Software notes

landscapemetrics: an open-source *R* tool to calculate landscape metrics

Maximilian H. K. Hesselbarth, Marco Sciaini, Kimberly A. With, Kerstin Wiegand and Jakub Nowosad

M. H. K. Hesselbarth (https://orcid.org/0000-0003-1125-9918) ✉ (maximilian.hesselbarth@uni-goettingen.de), M. Sciaini (https://orcid.org/0000-0002-3042-5435) and K. Wiegand (https://orcid.org/0000-0003-4854-0607), Dept of Ecosystem Modelling, Faculty of Forest Sciences and Forest Ecology, Univ. of Goettingen, Buesgenweg 4, DE-37077 Goettingen, Germany. – K. A. With (https://orcid.org/0000-0001-5570-1515), Laboratory for Landscape and Conservation Ecology, Division of Biology, Kansas State Univ., Manhattan, KS, USA. – J. Nowosad (https://orcid.org/0000-0002-1057-3721), Inst. of Geoecology and Geoinformation, Adam Mickiewicz Univ., Poznan, Poland.

Quantifying landscape characteristics and linking them to ecological processes is one of the central goals of landscape ecology. Landscape metrics are a widely used tool for the analysis of patch-based, discrete land-cover classes. Existing software to calculate landscape metrics has several constraints, such as being limited to a single platform, not being open-source or involving a complicated integration into large workflows. We present *landscapemetrics*, an open-source *R* package that overcomes many constraints of existing landscape metric software. The package includes an extensive collection of commonly used landscape metrics in a tidy workflow. To facilitate the integration into large workflows, *landscapemetrics* is based on a well-established spatial framework in *R*. This allows pre-processing of land-cover maps or further statistical analysis without importing and exporting the data from and to different software environments. Additionally, the package provides many utility functions to visualize, extract, and sample landscape metrics. Lastly, we provide building-blocks to motivate the development and integration of new metrics in the future. We demonstrate the usage and advantages of *landscapemetrics* by analysing the influence of different sampling schemes on the estimation of landscape metrics. In so doing, we demonstrate the many advantages of the package, especially its easy integration into large workflows. These new developments should help with the integration of landscape analysis in ecological research, given that ecologists are increasingly using *R* for the statistical analysis, modelling and visualization of spatial data.

Keywords: landscape analysis, landscape indices, landscape mosaic model, open-source software, *R* software, sampling design

Introduction

Understanding how landscape characteristics affect ecological processes and the spatial distribution of species and communities is central to ecology (Turner 1989, 2005, Kupfer 2012). Thereby, one major challenge is how to describe and quantify landscape characteristics (Turner 2005, Lausch et al. 2015). Typically, landscapes

are characterized as discrete patches of different land-cover classes (i.e. a landscape mosaic, Forman and Godron 1986, Forman 1995, Wiens 1995) which has several benefits. These include a straightforward application and communication (McGarigal et al. 2009, Lausch et al. 2015), especially in human-dominated landscapes where the distinction between different land-cover classes is rather clear-cut (Lausch et al. 2015). While other landscapes may be better described by a gradient-based description of landscape structure (McGarigal et al. 2009, Cushman et al. 2010), the landscape-mosaic model remains the dominant paradigm (Kupfer 2012, With 2019).

To quantify the composition (number and abundance) and configuration (spatial arrangement) of different land-cover classes, numerous landscape metrics have been developed and extensively applied to the analysis of landscape structure (Gustafson 1998, 2019, Uuemaa et al. 2013). Such landscape metrics are commonly used to facilitate comparisons among different landscapes; to quantify how landscapes change over time, especially in response to different types of disturbances or land-use pressures; and to investigate the relationship between landscape characteristics and other ecological patterns (Uuemaa et al. 2009). Some recent examples of these sorts of applications include studies of how landscape characteristics of temperate forests differ between years of low and high natural disturbance activities (Senf and Seidl 2018), how land-use intensity affects agricultural landscapes and associated biodiversity (Decaëns et al. 2018), and how distributional patterns of birds with different habitat affinities are related to landscape heterogeneity (Herrera et al. 2018).

One commonly used software to calculate landscape metrics is the stand-alone software *FRAGSTATS* (McGarigal et al. 2012). First published in 1995, *FRAGSTATS* was the first software to provide an extensive collection of landscape metrics, and subsequently, revolutionized landscape pattern analysis (Kupfer 2012, Gustafson 2019). However, ecologists are increasingly turning to *R* (Sciaini et al. 2018), a language originally developed for statistical computing (<www.r-project.org>). Nowadays, *R* is more and more used for the analysis, modelling and visualization of spatial data (Fletcher and Fortin 2018). One benefit of *R* is its active community that constantly develops software packages for specific tasks. The *R* package *SDMTools* (VanDerWal et al. 2019) includes a small subset of landscape metrics, but until now there is no comprehensive, dedicated *R* package to calculate landscape metrics. Therefore, the development of new software that facilitates the application of landscape metrics is still necessary to keep pace with the changing needs and expectations of ecologists wanting to perform landscape analysis.

Here, we present *landscapemetrics*, an extensive collection of widely used landscape metrics for the analysis of discrete land-cover maps, including the most commonly used metrics (Cushman et al. 2008, Schindler et al. 2008, Lustig et al. 2015), as well as some recent ones (e.g. joint entropy, Nowosad and Stepinski 2019). To demonstrate its application, we present an analysis of how different sampling

schemes influence the estimation of landscape metrics using neutral landscape models that vary in spatial autocorrelation by adopting a virtual ecologist approach (Zurell et al. 2010, Supplementary material Appendix 1 Fig. A1).

The *R* package *landscapemetrics*

The core of *landscapemetrics* comprises functions to calculate landscape metrics and uses raster data as input. Therefore, the package is mainly based on the well-established *raster* package (Hijmans 2019), but the use of next-generation frameworks is also possible (*stars* package, Pebesma 2019).

At present, *landscapemetrics* primarily includes the so-called *FRAGSTATS*-style metrics (Kupfer 2012, McGarigal et al. 2012), but other types of metrics are planned for future updates. The current software version includes metrics on all available levels, namely patch-, class- and landscape-level. Patch-level metrics describe every patch in a landscape (a patch being defined as contiguous cells belonging to the same land-cover class). Class-level metrics describe all patches belonging to a certain land-cover class. Lastly, landscape-level metrics describe the characteristics of the entire landscape (McGarigal et al. 2012). Additionally, landscape metrics can be classified according to the characteristics of the landscape they (conceptually) describe (McGarigal et al. 2012, Šímová and Gdulová 2012). *landscapemetrics* includes area and edge metrics, shape metrics, core area metrics, aggregation metrics, diversity metrics, as well as complexity metrics. For a full list of all metrics, see the package documentation (<https://r-spatialecology.github.io/landscapemetrics>).

Improvements over existing software tools

Though popular, *FRAGSTATS* has certain drawbacks (Table 1). As a stand-alone software, it requires data import to the software for integration into large workflows. If the resulting metrics are the basis for further analysis, they must then be exported to yet another program. Additionally, the analysis of several input layers or the use on high-performance clusters is rather laborious. Also, *FRAGSTATS* is not open-source software and only available for *Windows* operating systems. This can complicate transparency and reproducibility of the analysis workflow, and collaboration among researchers using different computing platforms.

Contrastingly, *R* is open-source and available for most common operating systems (including Windows, macOS and Linux). The existing *R* package *SDMTools* calculates a limited number of landscape metrics, and thereby overcomes some of the above-mentioned limitations of *FRAGSTATS*. But, as this package was primarily developed for species distribution modelling rather than landscape analysis, it cannot fully replace *FRAGSTATS* (Table 1). The use of the package for landscape analysis is rather cumbersome. To calculate patch-level metrics, the data must first be converted to a matrix and a loop through all land-cover classes implemented

Table 1. Main characteristics and features of *FRAGSTATS*, *SDMTools* and *landscapemetrics*. The available metric levels are abbreviated: p=patch level, c=class level, l=landscape level.

Characteristics	FRAGSTATS	SDMTools	landscapemetrics
open-source software	no	yes	yes
cross-platform compatibility	no	yes	yes
available metric levels	p, c, l	p, c	p, c, l
parametrization of metrics	yes	no	yes
tidy data format	no	no	yes
easy integration into workflows	no	no	yes
utility functions	sampling	no	various
restrictions due to numeric precision	yes[1]	no	no

[1] *FRAGSTATS* does not allow a cell resolution <0.005 map units of the input raster.

(Example 1). Furthermore, *SDMTools* does not permit a sub-selection of metrics and no further parameterization of metrics is possible. Additionally, the user must specify the cell resolution manually for all area- and distance-related calculations, introducing a possible error source. Also, the output format makes integration of the results into large workflows difficult. Lastly, to our knowledge, *SDMTools* is not actively developed anymore.

landscapemetrics provides an extensive collection of widely used landscape metrics for discrete land-cover maps, including most of the commonly used metrics used in landscape analysis (Cushman et al. 2008, Schindler et al. 2008, Lustig et al. 2015), as well as some recent ones (e.g. joint entropy, Nowosad and Stepinski 2019). Because *landscapemetrics* is written in the *R* programming language, it operates across operating platforms. Given the

```
library(dplyr)
library(landscapemetrics)
library(SDMTools)
library(raster)

#### SDMTools ####
result_patch <- list() # preallocate list
classes <- unique(landscape) # all present classes

# loop through all classes
for(i in seq_along(classes)) {
        landscape_matrix <- as.matrix(landscape) # convert to matrix
        landscape_matrix[landscape_matrix != i] <- 0 # binarize landscape
        landscape_matrix[landscape_matrix == i] <- 1 # binarize landscape
        ccl <- ConnCompLabel(landscape_matrix) # get patches
        result_patch[[i]] <- PatchStat(ccl, cellsize = 1) # patch metrics
}

result_patch <- bind_rows(result_patch, .id = "classID") # combine to one df

result_patch <- filter(result_patch, patchID != 0) # only present classes

area_SDM <- result_patch[, c(1, 2, 7)] # select only area
perim_SDM <- result_patch[, c(1, 2, 9)] # select only perimeter

#### landscapemetrics ####
area_lsm <- lsm_p_area(landscape) # calculate patch area
perim_lsm <- lsm_p_perim(landscape) # calculate patch perimeter
```

Example 1. Comparison of *SDMTools* and *landscapemetrics* workflows to calculate patch area and patch perimeter. *SDMTools* requires a loop through all present land-cover classes and a binarization of the input to 1 for the current class and 0 for all other classes. Also, the desired metrics must be filtered from the resulting data frame.

Table 2. Structure of the output table used for all metrics in *landscapemetrics*. The output is type stable, which simplifies integration into larger workflows.

Layer <integer>	Level <character>	Class <integer>	ID <integer>	Metric <character>	Value <double>
ID of landscape	Level of metric	ID of class; NA for landscape level	ID of patch; NA for class and landscape level	Abbreviation of metric	Value of metric

variety of *R* packages, this also enables the user to run the software easily in parallel or on high-performance clusters. To ensure integration into large workflows, *landscapemetrics* is based on a well-established spatial framework in *R* (mainly the *raster* package). This allows pre-processing of data, calculation of metrics, and further analysis of the results, all within the same software environment. This also facilitates its usage with other spatial objects in *R* (e.g. *sp* spatial points, Pebesma and Bivand 2005). Additionally, the use of raster data has advantages in having all required spatial information included and eliminating possible error sources, such as a mis-specified cell size. To simplify integration further, the output of all metric functions is tidy (sensu Wickham 2014) and type stable, meaning the returning data frame is identically structured regardless of the level or metric (Table 2). This facilitates reproducible workflows (Sandve et al. 2013). The package is open-source, which allows users to comprehend and improve upon existing metrics, as well as to contribute new functions. All functions were designed to calculate landscape metrics in a straightforward way (Example 1). Lastly, the package provides several utility functions (Table 3) to facilitate visualization, extraction, sampling and development of metrics.

Calculation of landscape metrics

The first step of every analysis should be a check if the input raster is suitable for *landscapemetrics* using `check_landscape()`. The function checks if the coordinate reference system is projected, if the cell units are in meters, if the classes are decoded as integer values, and if the number of different values is reasonable (in other words if discrete land-cover classes are present). In case the input is not or only partially suitable, a corresponding warning is produced. This means that a calculation of metrics is still possible, but some results must be interpreted with caution (e.g. area- and distance-related metrics).

To get an overview of all available metrics, *landscapemetrics* provides the function `list_lsm()`. It is possible to specify metrics by name, level, and/or type. Of course, all specifications can be combined. Also, rather than a data frame, it is possible to return a vector with function names. For example, the vector format makes it straightforward to get only the function names of all patch- and class-level aggregation metrics for later use by `list_lsm(level=c("patch", "class"), type="aggregation metric", simplify=TRUE)`.

All functions to calculate landscape metrics are consistently named in *landscapemetrics*. Functions to calculate a given metric have the prefix 'lsm_' followed by an abbreviation for the level ('p', 'c' or 'l' for patch-, class- and landscape-level, respectively) and lastly for the metric itself. For example, the class area for each land-cover class is calculated simply by `lsm_c_ca()`. All functions can handle several landscapes as input (either a list of RasterLayers or a RasterStack/RasterBrick).

Several metrics, regardless of the level, can easily be combined into one data frame because the resulting output is always an identically structured data frame (Table 2). Therefore, the results of the desired metrics can be combined using, for example, `rbind(lsm_p_area(x), lsm_l_ai(x))`. Because all output is tidy following widely accepted data science standards, further analysis of

Table 3. Overview of utility functions in the *landscapemetrics* package.

Area of application	Function name	Description
Visualization	show_patches()	Plot patches in the landscape
Visualization	show_cores()	Plot core areas in the landscape
Visualization	show_lsm()	Plot landscape filling cells with patch level metric value
Visualization	show_correlation()	Show correlation between metrics
Sampling	sample_lsm()	Sample metrics in a buffer around sample points
Sampling	extract_lsm()	Extract landscape metric of patches enclosing sample points
Sampling	window_lsm()	Moving window analysis
Building block	get_adjacencies()	Get class cell adjacencies
Building block	get_boundaries()	Get boundary cells of patches
Building block	get_circumscribingcircle()	Get diameter of the smallest circumscribing circle around patches
Building block	get_nearestneighbour()	Get minimum Euclidean distance between classes
Building block	get_patches()	Patch delineation
Various	check_landscape()	Check if input fulfils package requirements
Various	list_lsm()	List all available metrics
Various	spatialize_lsm()	Assign patch metric to each cell

the resulting data frame is possible without laborious data import/export or formatting (see Use case).

Many functions provide additional parametrization, such as the edge depth or the cell neighbourhood rule for patch delineation. For example, to change the patch delineation rule for the patch area, only the argument `directions` must be changed from its default using the 'queen's case' (eight neighbouring cells) to the 'rook's case' (four neighbouring cells), i.e. `lsm_p_area(x, directions=4)`. Of course, all arguments are consistently named across metrics.

The `calculate_lsm()` wrapper can be used to calculate several metrics simultaneously. Similar to `list_lsm()`, this wrapper allows specifying a subset of metrics using the name, level, and/or type, such as `calculate_lsm(x, level="landscape", type="diversity metric")`. Alternatively, a vector (e.g. previously created using `list_lsm()`) with function names can be provided as `what`-argument. The returning data frame is identical to the output of all single metric functions.

Utility functions

An additional advantage of *landscapemetrics* over existing software tools for landscape analysis lies in its utility functions (Table 3). These functions are designed to facilitate the application, visualization, extraction, sampling and development of landscape metrics.

Visualization functions, which help to understand and communicate metrics, start with the prefix 'show_' followed by the subject to visualize. It is possible to include either all classes in one plot (`class="global"`), all classes but each plotted separately (`class="all"`) or just selected classes (`class=c(1, 3)`). Patches in a landscape can be visualized by `show_patches()` (Fig. 1B), or to visualize only the core area, there is `show_cores()` (Fig. 1C). Additionally, patches can be filled with the value of any patch level metric, such as the patch area using `show_lsm(x, what="lsm_p_area")`. It is also possible the get the result as a RasterLayer, using `spatialize_lsm()`.

In the returning RasterLayer, each cell has the value of the corresponding patch for any chosen metric. Correlations between metrics can be problematic (Cushman et al. 2008, Schindler et al. 2008, Nowosad and Stepinski 2018b) and the selection of mainly uncorrelated metrics can be a challenge. Providing a data frame with metric results, `show_correlations()` returns a correlation matrix plot.

There are several functions to sample landscape metrics. Sample locations can be provided either as a matrix including x- and y-coordinates or as *sp*-objects and for all sampling functions, the metrics can be specified similar to `list_lsm()`. The function `extract_lsm()` returns the patch-level metric values of each patch in which sample points are located. To calculate metrics in a buffer around sample points, `sample_lsm()` can be used, allowing to specify the shape (circle, square or rectangle) and the area of buffers around sample points, and then calculates the specified metrics. Landscape metrics are known to be scale dependent (Lausch and Herzog 2002, Wu 2004, Lustig et al. 2015) and using a moving window can be an approach to deal with this (Su et al. 2011). The moving window assigns to each focal cell in the landscape the metric value of its local neighbourhood specified by a neighbourhood matrix (McGarigal et al. 2012). The resulting raster describes the landscape in regard to the local variability of the chosen metric (Hagen-Zanker 2016). Within *landscapemetrics*, a moving window approach can be applied by using `window_lsm()`. The local neighborhood can be specified using, for example, `window <- matrix(1, nrow=5, ncol=5)`, followed by `window_lsm(x, window=window, what=c("lsm_l_pr", "lsm_l_joinent"))`.

Lastly, *landscapemetrics* provides several building-blocks to develop and contribute new metrics. These functions all start with the prefix 'get_' and are mainly computationally fast *C/C++* implementations of common raster procedures. Of these, `get_patches()` is the most fundamental, as it returns all connected cells as patches and is used in most metrics. Another useful function is `get_adjacencies()`, which returns the adjacency matrix of all cells, and is easier to use, faster and more memory efficient than its equivalent
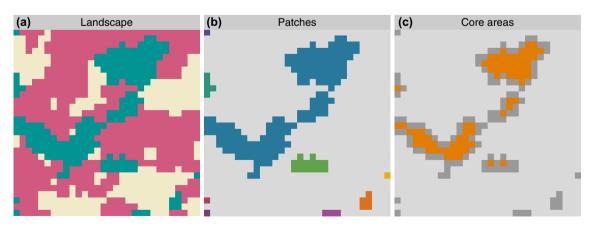

Figure 1. Visualization of an example landscape (a) using the utility functions `show_patches()` (b) and `show_cores()` (c) for a selected class.

```
library(landscapemetrics)
library(dplyr)
library(bench)

#### raster ####
adj_raster <- function(x) {
      # get cell ids of neighboring cells
      adjacencies <- raster::adjacent(x, cells = 1:raster::ncell(x))
      # table of values of neighboring cells
      table(x[adjacencies[, 1]], x[adjacencies[, 2]])
}

adj_raster(landscape)

#### landscapemetrics ####
get_adjacencies(landscape)

#### benchmark of both options ####
mark(adj_raster(landscape),
        get_adjacencies(landscape),
        iterations = 10000, check = FALSE)

# A tibble: 2 x 3
    expression                        mean          mem_alloc
    <chr>                          <bch:tm>          <bch:byt>
1 adj_raster(landscape)             8900 µs           1470 KB
2 get_adjacencies(landscape)         562 µs              6 KB
```

Example 2. Comparison between *raster* and *landscapemetrics* to get the cell adjacency matrix of a raster. Not only is the *landscapemetrics* solution much easier, but it is also computationally faster and more memory efficient. The benchmark was done on *Windows* 10 (32 GB RAM, Intel i7 core, 3.4 GHz), using *R* ver. 3.5.1 and 10 000 iterations of each function.

in the *raster* package (Example 2). Finally, `get_nearest-neigbhour()` returns the minimum Euclidean distance between patches of the same class, and `get_circum-scribingcircle()` returns the diameter of the smallest circumscribing circle around each patch.

Use case

Here, we aim to demonstrate some of the advantages of the package, especially its easy integration into large workflows. To this end, we apply *landscapemetrics* to analyse the effect of different sampling schemes on the estimation of landscape metrics. All code to reproduce the use case can be found at <https://zenodo.org/record/2597976>.

Although the collection, processing and analysis of spatial data across landscape and regional scales has become commonplace, ecologists still need to subsample data from a larger landscape. Possible reasons are atmospheric conditions, logistical or budget constraints, or the use of unmanned aerial vehicles, resulting in high-resolution imagery but with limited extent (Getzin et al. 2012). Even though landscape metrics are known to be sensitive to various scaling issues

(Lausch and Herzog 2002, Wu 2004, Lustig et al. 2015), the quality of the sample mean as an estimator has only been investigated for a subset of metrics and specific sampling schemes (but see Ramezani et al. 2010, Ramezani and Holm 2011, Hassett et al. 2012).

We used the virtual ecologist approach (Zurell et al. 2010, Supplementary material Appendix 1 Fig. A1) which can be summarised in four major steps: 1) a virtual ecological simulation model of an ecosystem (or landscape, in this case), 2) a virtual sampling process, sampling data from the virtual ecosystem or landscape, 3) analyses of the sampled data and 4) an evaluation of the results against the true value for the full virtual ecosystem or landscape (Zurell et al. 2010). Following this approach, we first simulated neutral landscapes (500 × 500 cells) containing five classes (relative proportion of 20% each) with either low, medium or high spatial autocorrelation, respectively (*NLMR* package, Sciaini et al. 2018). For each landscape, we calculated all available landscape-level metrics that were invariant to the absolute plot area (Supplementary material Appendix 1 Table A1). Within each landscape, we sampled data using all 54 possible combinations of plot size, landscape area sampled, plot shape
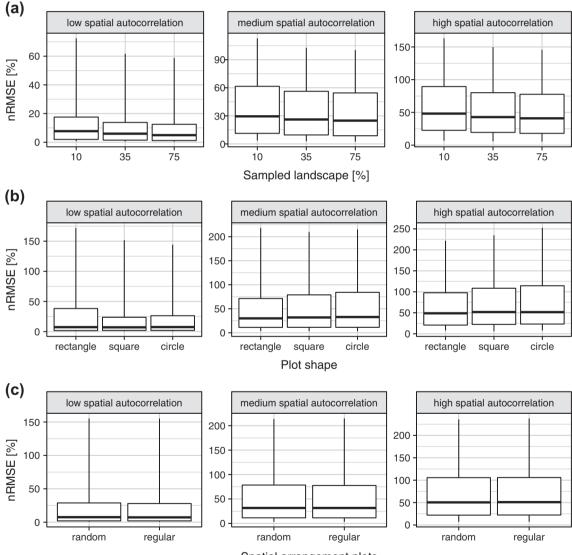
Table 4. Components of the sampling scheme. All 54 possible unique combinations were used for the analyses.

Size (cells)			Sampled landscape (%)			Shape			Arrangement	
2500	7500	20 000	~10	~35	~75	Rectangle	Square	Circle	Random	Regular

and spatial arrangement (Table 4). We used the sample mean to estimate the landscape properties for the whole landscapes and evaluated the estimated metrics against the true metrics using the root–mean–square error (RMSE, Hyndman and Koehler 2006) normalized by the mean

as $nRMSE = \sqrt{\dfrac{\text{Var}(\overline{x})}{n-1} + (\overline{x} - \mu)^2} / \overline{x}$.

The metrics 'area_cv', 'area_sd', 'core_cv', 'core_sd', 'lsi' and 'mesh' were estimated with a *nRMSE* >125% for all sampling schemes and spatial autocorrelations, and were thus removed from further analyses. In general, we found that the accuracy and precision of the estimator decreased with increasing spatial autocorrelation (Fig. 2). Across all metrics and spatial autocorrelations, the median *nRMSE* decreased slightly as the area sampled increased (Fig. 2a). The median

Figure 2. Normalized root–mean–square error (*nRMSE*) for different sampling schemes. The *nRMSE* is summarized for all metrics and sampling schemes components not present on the corresponding x-axis. For (a) the sampled landscape is increased, for (b) different sampling plot shapes and for (c) different spatial arrangements of sampling plots are used. The solid lines represent the median, the boxes the middle 50% of the data and the whiskers include 1–99% of the data. The y-scales differ among panels to highlight differences between the sampling schemes.

nRMSE across all metrics and spatial autocorrelations was similar among plot shapes, but the extreme deviations slightly increased for square and circular plots with increasing spatial autocorrelation (Fig. 2b). Finally, the spatial arrangement of plots had no clear influence on accuracy and precision of the estimator (Fig. 2c).

We found that most landscape metrics were able to provide a fairly accurate and precise assessment of landscape structure for landscapes with a low spatial autocorrelation, likely because individual plots captured more of the inherent spatial heterogeneity present within the overall landscape (Wiens 1989, Hassett et al. 2012). Accuracy and precision decreased with increasing spatial autocorrelation, likely due to increasing between-plot variability (Hassett et al. 2012). Accuracy and precision of the estimator increased as the area sampled within the landscape increased because estimators increasingly converged on the true landscape value. Although this has been found by others (Ramezani and Holm 2011), the influence was smaller than expected, perhaps because the same region of the landscape might have been resampled due to overlapping sample plots. There were no clear differences between the three plot shapes investigated (rectangular, circular, square). When spatial autocorrelation was low, the 'salt and pepper' properties of the landscapes were adequately captured by all plot shapes. Contrastingly, with increasing spatial autocorrelation, rectangular plots provided a slightly more accurate and precise estimation of landscape properties than did circular and square plots. Most likely, rectangular plots captured more spatial heterogeneity. A similar effect can be observed for species richness counts, where more species can be found in elongated plots (Güler et al. 2016). Both regular and random arrangements of sampling plots were able to capture landscape properties similarly well. This is not surprising for landscapes with a low spatial autocorrelation, because the finer scale of heterogeneity means all sampling distributions should perform similarly. However, in landscapes with high spatial autocorrelation, a regular sampling array should better capture landscape structure than a random sampling scheme, given the coarser scale of heterogeneity (Ramezani et al. 2010).

Discussion

The use case demonstrated many advantages of the *landscapemetrics* package, especially the easy integration into large analysis workflows without the need of importing/exporting to or from different software environments. This also facilitates usage in combination with other spatial *R* packages, such as *raster*, *sp* or *NLMR*.

Although the aimless calculation of landscape metrics is not recommended (Gustafson 2019), in cases where many metrics need to be calculated, *landscapemetrics* can calculate these with ease either in parallel or on high-performance clusters. This lessens one disadvantage of *landscapemetrics*

being computationally slower for the calculation of some landscape metrics compared to *FRAGSTATS* and *SDMTools*. The use of *landscapemetrics* in parallel computing or on high-performance clusters can also be a huge advantage for studies where only a small number of metrics need to be calculated, but for many landscapes, as in the use case demonstrated here with its large factorial design (see Use case).

The type-stable and tidy output also allows users to process results without much data formatting, using data manipulation packages such as *dplyr* (Wickham et al. 2019), and furthermore, to produce figures using plotting packages such as *ggplot2* (Wickham 2016).

Finally, being open-source and hosted on *GitHub* improves the transparency of the package and users can easily file bug reports to ensure a rapid fix. This can also lead to the development of new functions when requested by users on *GitHub*, as evidenced by several new functions that have been implemented since the first release of the *landscapemetrics* package.

Summary

landscapemetrics is the first *R* package that allows calculation of most of the commonly employed landscape metrics found in the ecological literature (Cushman et al. 2008, Schindler et al. 2008, Lustig et al. 2015). Along with the package, a dedicated website introduces the basic concepts and usage of *landscapemetrics* (<https://r-spatialecology.github.io/landscapemetrics/>).

Many characteristics, such as including a large set of landscape metrics, working across platforms, being open-source, and the ability to analyse a comprehensive variety of spatial data within larger workflows, are all advantages of the package. The included utility functions that permit the visualization, extraction, sampling and development of metrics, provide additional benefits over existing software. We hope this helps integrate *landscapemetrics* more easily into larger workflows, enhances transparency and reproducibility, and simplifies landscape analyses in ecological investigations.

Data deposition

Data are available from the Zenodo Digital Repository: <https://zenodo.org/record/2597976> (Hesselbarth et al. 2019).

Software availability

landscapemetrics is available on the comprehensive *R* archive network (*CRAN*): <https://CRAN.R-project.org/package=landscapemetrics> and is also hosted on *GitHub*: <https://www.github.com/r-spatialecology/landscapemetrics>. *landscapemetrics* is distributed under GNU Public License ver. 3 (GPLv3).

To cite *landscapemetrics* or acknowledge its use, cite this software note as follows, substituting the version of the application that you used for 'version 0':

References

Cushman, S. A. et al. 2008. Parsimony in landscape metrics: strength, universality and consistency. – Ecol. Indic. 8: 691–703.

Cushman, S. A. et al. 2010. The gradient paradigm: a conceptual and analytical framework for landscape ecology. – In: Spatial complexity, informatics and wildlife conservation. Springer, pp. 83–108.

Decaëns, T. et al. 2018. Biodiversity loss along a gradient of deforestation in Amazonian agricultural landscapes. – Conserv. Biol. 32: 1380–1391.

Fletcher, R. and Fortin, M.-J. 2018. Spatial ecology and conservation modeling. Applications with R. – Springer.

Forman, R. T. T. 1995. Land mosaics: the ecology of landscapes and regions. – Cambridge Univ. Press.

Forman, R. T. T. and Godron, M. 1986. Landscape ecology. – Wiley.

Getzin, S. et al. 2012. Assessing biodiversity in forests using very high-resolution images and unmanned aerial vehicles. – Methods Ecol. Evol. 3: 397–404.

Güler, B. et al. 2016. How plot shape and spatial arrangement affect plant species richness counts: implications for sampling design and rarefaction analyses. – J. Veg. Sci. 27: 692–703.

Gustafson, E. J. 1998. Quantifying landscape spatial pattern: what is the state of the art? – Ecosystems 1: 143–156.

Gustafson, E. J. 2019. How has the state-of-the-art for quantification of landscape pattern advanced in the twenty-first century? – Landscape Ecol. 34: 1–8.

Hagen-Zanker, A. 2016. A computational framework for generalized moving windows and its application to landscape pattern analysis. – Int. J. Appl. Earth Obs. Geoinf. 44: 205–216.

Hassett, E. M. et al. 2012. Estimating landscape pattern metrics from a sample of land cover. – Landscape Ecol. 27: 133–149.

Herrera, J. M. et al. 2018. A distribution-oriented approach to support landscape connectivity for ecologically distinct bird species. – PLoS One 13(4): e0194848.

Hesselbarth, M. H. K. et al. 2019. Data from: *landscapemetrics*: an open-source *R* tool to calculate landscape metrics. – Zenodo Digital Repository <https://zenodo.org/record/2597976>

Hijmans, R. J. 2019. raster: geographic data analysis and modeling. – R package ver. 2.9-5. <https://cran.r-project.org/package=raster>.

Hyndman, R. J. and Koehler, A. B. 2006. Another look at measures of forecast accuracy. – Int. J. Forecast. 22: 679–688.

Kupfer, J. A. 2012. Landscape ecology and biogeography: rethinking landscape metrics in a post-FRAGSTATS landscape. – Prog. Phys. Geogr. 36: 400–420.

Lausch, A. and Herzog, F. 2002. Applicability of landscape metrics for the monitoring of landscape change: Issues of scale, resolution and interpretability. – Ecol. Indic. 2: 3–15.

Lausch, A. et al. 2015. Understanding and quantifying landscape structure – a review on relevant process characteristics, data models and landscape metrics. – Ecol. Model. 295: 31–41.

Lustig, A. et al. 2015. Towards more predictable and consistent landscape metrics across spatial scales. – Ecol. Indic. 57: 11–21.

McGarigal, K. et al. 2009. Surface metrics: an alternative to patch metrics for the quantification of landscape structure. – Landscape Ecol. 24: 433–450.

McGarigal, K. et al. 2012. FRAGSTATS v4: spatial pattern analysis program for categorical and continuous maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. <http://www.umass.edu/landeco/research/fragstats/fragstats.html>.

Nowosad, J. and Stepinski, T. F. 2018. Global inventory of landscape patterns and latent variables of landscape spatial configuration. – Ecol. Indic. 89: 159–167.

Nowosad, J. and Stepinski, T. F. 2019. Information theory as a consistent framework for quantification and classification of landscape patterns. – Landscape Ecol. in press.

Pebesma, E. J. 2019. stars: scalable, spatiotemporal tidy arrays for R. – R package ver. 0.3-1. <https://CRAN.R-project.org/package=stars>.

Pebesma, E. J. and Bivand, R. S. 2005. Classes and methods for spatial data in R. – R News 5: 9–13.

Ramezani, H. and Holm, S. 2011. Sample based estimation of landscape metrics; accuracy of line intersect sampling for estimating edge density and Shannon's diversity index. – Environ. Ecol. Stat. 18: 109–130.

Ramezani, H. et al. 2010. Monitoring landscape metrics by point sampling: accuracy in estimating Shannon's diversity and edge density. – Environ. Monit. Assess. 164: 403–421.

Sandve, G. K. et al. 2013. Ten simple rules for reproducible computational research. – PLoS Comput. Biol. 9: e1003285.

Schindler, S. et al. 2008. Towards a core set of landscape metrics for biodiversity assessments: a case study from Dadia National Park, Greece. – Ecol. Indic. 8: 502–514.

Sciaini, M. et al. 2018. NLMR and landscapetools: an integrated environment for simulating and modifying neutral landscape models in R. – Methods Ecol. Evol. 9: 2240–2248.

Senf, C. and Seidl, R. 2018. Natural disturbances are spatially diverse but temporally synchronized across temperate forest landscapes in Europe. – Global Change Biol. 24: 1201–1211.

Šímová, P. and Gdulová, K. 2012. Landscape indices behavior: a review of scale effects. – Appl. Geogr. 34: 385–394.

Su, S. et al. 2011. Transformation of agricultural landscapes under rapid urbanization: a threat to sustainability in Hang-Jia-Hu region, China. – Appl. Geogr. 31: 439–449.

Turner, M. G. 1989. Landscape ecology: the effect of pattern on process. – Annu. Rev. Ecol. Syst. 20: 171–197.

Turner, M. G. 2005. Landscape ecology: what is the state of the science? – Annu. Rev. Ecol. Evol. Syst. 36: 319–344.

Uuemaa, E. et al. 2009. Landscape metrics and indices: an overview of their use in landscape research. – Living Rev. Landscape Res. 3: 1–28.

Uuemaa, E. et al. 2013. Trends in the use of landscape spatial metrics as landscape indicators: a review. – Ecol. Indic. 28: 100–106.

VanDerWal, J. et al. 2019. SDMTools: species distribution modelling tools: tools for processing data associated with species distribution modelling exercises. – R package ver. 1.1-221.1. <https://CRAN.R-project.org/package=SDMTools>.

Wickham, H. 2014. Tidy data. – J. Stat. Softw. 59: 1–23.

Wickham, H. 2016. ggplot2: elegant graphics for data analysis. – Springer.

Wickham, H. et al. 2019. dplyr: a grammar of data manipulation. – R package ver. 0.8.1. <https://CRAN.R-project.org/package=dplyr>.

Wiens, J. A. 1989. Spatial scaling in ecology. – Funct. Ecol. 3: 385–397.

Wiens, J. A. 1995. Landscape mosaics and ecological theory. – In: Hansson, L. et al. (eds), Mosaic landscapes and ecological processes. Chapman and Hall, pp. 1–26.

With, K. A. 2019. Essentials of landscape ecology. – Oxford Univ. Press.

Wu, J. 2004. Effects of changing scale on landscape pattern analysis: scaling relations. – Landscape Ecol. 19: 125–138.

Zurell, D. et al. 2010. The virtual ecologist approach: simulating data and observers. – Oikos 119: 622–635.

Supplementary material (available online as Appendix ecog-04617 at <www.ecography.org/appendix/ecog-04617>). Appendix 1.